# Trabajo Practico N°1

## Inteligencia Artificial

Prieto Julian 45065709

## 1)

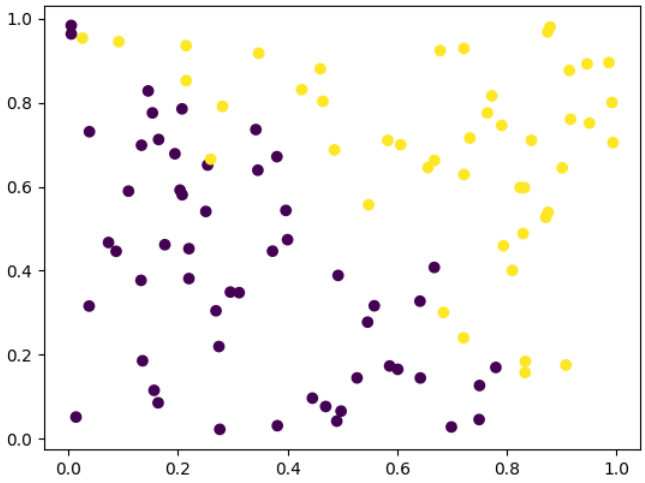
### A

La diferencia mas notable al entrenar ambos conjuntos de datos es que el conjunto A llega a la convergencia, es decir, alcanza el maximo error elegido, de forma mucho mas rapida que el conjunto de datos B. Este ultimo tarda muchisimo mas tiempo en llegar a dicha convergencia.

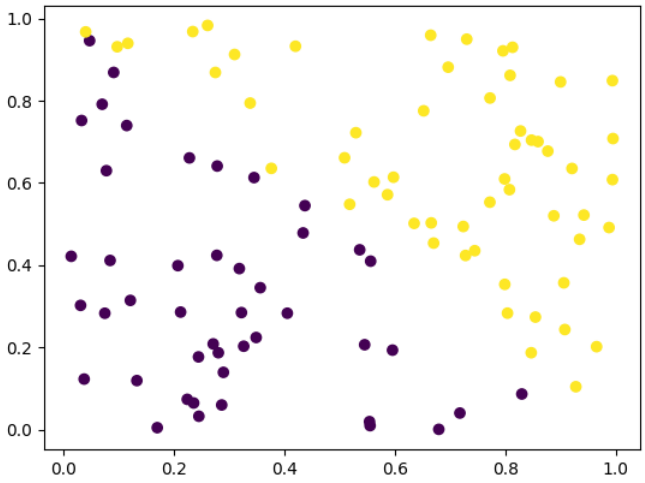
### B

Habiendo hecho pruebas con los datesets, se logró poder visualizar ambas clasificaciones en una gráficos de puntos.

Dataset A




Dataset B




Como se puede observar, el dataset B el linealmente separable, lo que indica que se podria lograr una clasificion perfecta con un borde de decision lineal. Teniendo en cuenta esto, es posible que la diferencia entre datasets y la falta de convergencia del B sea a causa de esto.

## C

I. El learning rate es un hiperparámetro muy importante en algoritmos de aprendizaje automático, incluyendo modelos de regresión logística. Afecta la velocidad y la estabilidad del proceso de entrenamiento de un modelo de regresión logística.

El valor del learning rate determina qué tan rápido convergerá el algoritmo de optimización hacia los valores óptimos de los parámetros del modelo. Si el learning rate es demasiado pequeño, el modelo puede converger lentamente y requerir más iteraciones para alcanzar una solución óptima. Por otro lado, si el learning rate es demasiado grande, el algoritmo podría no converger o incluso divergir, lo que significa que los parámetros nunca alcanzarán una solución estable.

Por otra parte, un learning rate adecuado es crucial para la estabilidad del entrenamiento. Si el learning rate es demasiado alto, el modelo puede oscilar alrededor del mínimo óptimo o incluso divergir, lo que se conoce como "exploding gradients". Si el learning rate es demasiado pequeño, el modelo puede quedarse atascado en mínimos locales o tomar mucho tiempo en converger.

Se probó varios learning rates (tanto valores altos como bajos), pero no se logró mejorar el rendimiento del entrenamiento del algoritmo. Debido a esto, se entiende que no es la solucion que se busca para este problema.

---

II. Se probó disminuyendo el learning rate a cada iteracion **learning_rate = 1 / i^2** y se logró que ambos datasets convergan. Sin embargo, los dos tardan entre 9 y 14 millones de iteraciones en hacerlo.

---

III. El escalado de datos es un paso importante en el preprocesamiento de datos cuando se trabaja con modelos de regresión logística y otros algoritmos de aprendizaje automático.

La regresión logística utiliza técnicas de optimización, como el descenso de gradiente, para encontrar los parámetros óptimos del modelo. El escalado de datos puede ayudar a que estos algoritmos de optimización converjan más rápido y de manera más estable. Si las características no están en la misma escala, algunas de estas pueden dominar sobre otras en términos de contribución a la función de coste, lo que puede hacer que el proceso de optimización sea más lento.

Ademas, el escalado de datos también ayuda a igualar la importancia relativa de las diferentes características. Si algunas características tienen rangos de valores mucho mayores que otras, la regresión logística podría dar más peso a las características con valores más grandes. Esto puede ser problemático si todas las características son igualmente importantes para la predicción.

Realizando las pruebas necesarias, parece ser que escalando las X antes de realizar el entrenamiento no tuviera efecto alguno en la convergencia de los algoritmos. El dataset B sigue sin converger.

---

IV. La aplicación de un término de regularización al entrenamiento de un modelo de regresión logística puede tener un impacto significativo en la capacidad del modelo para generalizar y evitar el sobreajuste. La regresión logística es propensa al sobreajuste cuando se entrena con conjuntos de datos ruidosos o cuando se ajusta de manera demasiado ajustada a los datos de entrenamiento. Aquí se explican cómo afecta la regularización al entrenamiento de un modelo de regresión logística:

La regularización, ya sea L1 (Lasso) o L2 (Ridge), introduce un término adicional en la función de costo que penaliza los valores grandes de los coeficientes del modelo. Esto evita que los coeficientes se vuelvan muy grandes en magnitud, lo que puede llevar al sobreajuste. Un coeficiente grande significa que el modelo está dando demasiada importancia a una característica específica y, por lo tanto, está más sujeto a adaptarse a ruido en los datos de entrenamiento. La regularización ayuda a suavizar los coeficientes y, por lo tanto, a evitar el sobreajuste.

Agregando un termino de regularizacion a la funcion de costo, el gradiente nos queda con un termino de un valor lambda sumando unicamente. Haciendo esto, y probando con varios valores de lambda, se logro que el dataset B llegue a converger (utilizando $\lambda = 0.0225$), sin modificar el comportamiento del entrenamiento del dataset A.

---

V. Se probó sumar ruido gaussiano a los datos, tanto a los features como a los labels. Cabe destacar, que cada iteracion es un caso aparte, ya que los valores generados son pseudo-aleatorios, por lo que los resultados pueden variar.

Para el caso de las features, los modelos tuvieron un gran desempeño en el entrenamiento, logrando converger en menos de 10 mil iteraciones en casi todos los intentos. Ademas, generando un borde de decision acorde. Por otro lado, agregando ruido a los labels no se logra ninguna mejoría en el entrenamiento. El dataset B continúa sin poder llegar a la convergencia.

## D

No, las SVM en realidad son menos propensas a sufrir problemas de convergencia ya que son menos sensibles a la elección de los hiperparametros o a la calidad de los datos en comparación de la regresión logística. La función de pérdida de Hinge, utilizada por las SVM, están diseñadas para maximizar el margen entre las clases, lo que significa que se enfoca en los datos cercanos al borde de decisión que están clasificados incorrectamente. Mas especificamente, se centran en maximizar el margen funcional, es decir, la distancia entre un punto de datos y el hiperplano ponderada por la magnitud del vector de pesos del modelo. Que por ejemplo, se diferencia del margen geometrico, el cual se basa en la magnitud del vector de pesos sin considerar las etiquetas de clase.

## 2)

### A

Partiendo de el gradiente de la funcion de costo,

$$\nabla J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[ h_\theta(x^{(i)}) - y^{(i)} \right] x^{(i)}$$

igualamos a 0 y despejamos.

$$\frac{1}{m} \sum_{i=1}^{m} \left[ h_\theta(x^{(i)}) - y^{(i)} \right] x^{(i)} = 0$$

$$\sum_{i=1}^{m} h_\theta(x^{(i)}) - \sum_{i=1}^{m} y^{(i)} = 0$$

$$\sum_{i=1}^{m} h_\theta(x^{(i)}) = \sum_{i=1}^{m} y^{(i)}$$

$$\sum_{i=1}^{m} p(y^i = 1 | x^i, \theta) = \sum_{i=1}^{m} I\{y^i = 1\}$$

$$\boxed{\frac{\sum_{i=1}^{m} p(y^i = 1 | x^i, \theta)}{|\{i \in I_{a,b}\}} = \frac{\sum_{i=1}^{m} I\{y^i = 1\}}{|\{i \in I_{a,b}\}}}$$

### B

Tener un modelo perfectamente calibrado no implica conseguir una precision perfecta ya que aún así, el modelo podría estar prediciendo erroneamente debido a un umbral de desicion incorrecto, haciendo que clasifique algunos ejemplos como falsos negativos o como falsos positivos. Entonces, por mas que el promedio de de clasificaciones sea perfecto, el modelo se puede estar equivocando en qué casos clasifica como positivos y cuales negativos. Por otro lado, que el modelo tenga una precision perfecta no quiere decir que tenga una calibración perfecta. La precision esta enfocada en la exactitud de las predicciones de clase en base a verdaderos positivos y verdaderos negativos, lo que no afecta necesariamente a la calibracion del mismo ya que las probabilidades estimadas pueden llegar a ser inexactas en terminos de reflejar la probabilidad real.

En resumen, la calibración y la precisión son dos aspectos diferentes de la evaluación de un modelo de clasificación binaria. Un modelo puede estar perfectamente calibrado sin lograr precisión perfecta y viceversa. La calibración se refiere a la precisión de las probabilidades estimadas, mientras que la precisión se refiere a la precisión en la clasificación de ejemplos en términos de verdaderos positivos y verdaderos negativos.

### C

La regularizacion L2 o de Ridge es, de manera sencilla, una tecnica utilizada comunmente en la regresion logistica la cual se utiliza para prevenir sobreajustes en los modelos y aumentar el rendimiento de los mismos. Realizar una regularizacion a un modelo puede afectar la calibracion del mismo tanto de manera positiva como de manera negativa, dependiendo del contexto y de la eleccion del hiperparametro de regularizacion Lambda.

Por un lado, la regularizacion, como se menciono anteriormente, puede ayudar a disminuir el riesgo de sobreajuste del modelo predictivo haciendo que este generalice mejor y por lo tanto mejore sus predicciones en datos que nunca vió. Por otro lado, la regularizacion tambien puede llegar a simplificar de mas el modelo, llevando a una disminucion de la 'sensibilidad' del modelo a las caracteristicas de los datos y consiguiendo que este no se vea afectado por nuevos datos.

Por ultimo, algo muy importante de la regularizacion en la eleccion del hipreparametro Lambda. El efecto que tenga esta tecnica en el modelo depende mucho del valor de este parametro, ya que estableciendolo con un valor muy pequeño puede causar que el modelo se ajuste demasiado a los datos, provocando sobreajuste u overfitting. Pero, dandole un valor muy grande a Lambda aumentara la simpleza del modelo haciendo que este no pueda capturar la complejidad de los datos. En resumidas cuentas, la eleccion del valor del hiperparametro Lambda es de mucha importancia para la regularizacion y debe ser elegido teniendo en cuenta los datos, el modelo elegido y muchas otras cosas para conseguir un efecto positivo en el mismo.

## 3)

### A

Partiendo de que

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\, p(\theta|x,y)$$

Aplicamos la regla de Bayes

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\, \frac{p(y|x,\theta).p(\theta)}{p(y|x)}$$

Como $p(y|x)$ no depende 0, no lo tomamos en cuenta.

$$\boxed{\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\, p(y|x,\theta).p(\theta)}$$

## B

Sabiendo que

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\, p(y|x,\theta).p(\theta)$$

y que

$$p(\theta) = exp\left(\frac{-1}{2\eta^2}||\theta||_2^2\right)$$

Podemos escribir a $\theta_{MAP}$ como:

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\, p(y|x,\theta).exp\left(\frac{-1}{2\eta^2}||\theta||_2^2\right)$$

Aplicando logaritmo...

$$\log\theta_{MAP} = \log\left(\underset{\theta}{\operatorname{argmax}}\, p(y|x,\theta).exp\left(\frac{-1}{2\eta^2}||\theta||_2^2\right)\right)$$

$$\log\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\, \log(p(y|x,\theta)) - \frac{1}{2\eta^2}||\theta||_2^2$$

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmin}} - \log p(y|x,\theta) + \frac{1}{2\eta^2}||\theta||_2^2$$

Si reemplazamos $\lambda$ por $\frac{1}{2\eta^2}$:

$$\boxed{\theta_{MAP} = \underset{\theta}{\operatorname{argmin}} \left[-\log p(y|x,\theta) + \lambda||\theta||_2^2\right]}$$

## C

Teniedo en cuenta que

$$p(y|x,\theta) = p(y|x,\theta).p(\theta)$$

y dado que $\theta \sim N(\theta, \eta^2 I)$:

$$p(y|x,\theta) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}.\exp\left[\frac{-(y-\theta^T x)^2}{2\sigma^2}\right]\right).\left(\frac{1}{\sqrt{2\pi\sigma^2}}.\exp\left[\frac{-1}{2\eta^2}\theta^T\theta\right]\right)$$

Aplicando logaritmo natural en ambos lados,

$$\ln p(y|x,\theta) = ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}.\exp\left[\frac{-(y-\theta^T x)^2}{2\sigma^2}\right]\right).\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}.\exp\left[\frac{-1}{2\eta^2}\theta^T\theta\right]\right)$$

$$\ln p(y|x,\theta) = -\frac{1}{2}\ln\left(2\pi\sigma^2\right) - \frac{(y-\theta^T x)^2}{2\sigma^2} - \frac{1}{2}\ln\left(2\pi\sigma^2\right) - \frac{\theta^T\theta}{2\eta^2}$$

Sacando los valores constantes, nos queda:

$$\ln p(y|x,\theta) = -\frac{(y-\theta^T x)^2}{2\sigma^2} - \frac{\theta^T\theta}{2\eta^2}$$

Ahora derivamos con respecto a $\theta$:

$$\nabla \ln p(y|x,\theta) = \frac{x(y-\theta^T x)}{\sigma^2} - \frac{\theta}{\eta^2}$$

Igualamos a 0 y despejamos para $\theta$:

$$\frac{x(y-\theta^T x)}{\sigma^2} - \frac{\theta}{\eta^2} = 0$$

$$\frac{x(y-\theta^T x)}{\sigma^2} = \frac{\theta}{\eta^2}$$

$$\frac{x(y-\theta^T x)\eta^2}{\sigma^2} = \theta$$

Como tanto $\eta^2$ y $\sigma^2$ son la varianza, se cancelan. Y así conseguimos una expresion cerrada de $\theta_{MAP}$

$$\boxed{x(y-\theta^T x) = \theta}$$

## 4)

### A

Realizando el procedimiento descrito, se logro el siguiente resultado:

Imagen Original: 768.14 KB

Imagen Comprimida: 97.82 KB

Haz doble clic (o ingresa) para editar

### B

El factor de compresion se calculo en base a los bits de cada imagen. Para el calculo de los bits de la imagen original, se calculo la resolucion de la imagen y se lo multiplica por la cantidad de bits por pixel (y por canal de color), es decir, 24(8x3). Luego, los bits de la imagen comprimida es la resolucion por el logaritmo en base 2 de 16 (lo que calcula los bits por pixel teniendo en cuenta esa cantidad de colores), es decir, 4 bits por pixel. Al hacer esta cuenta, se divide la cantidad de bits de la imagen original por la cantidad de bits de la imagen comprimida, y se llega a que el factor de compresion al disminuir la cantidad de colores a 16 en total es de 6.